9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)
and
Air
21 - 23 September 2009, Hilton Head, South Carolina

AIAA 2009-7057

A Method of Optimally Combining Sectors

Michael C. Drew*

University Affiliated Research Center, Moffett Field, CA, 94035, USA

This paper describes a procedure for combining under-utilized sectors so that controllers can be allocated to the busiest sectors at the busiest times. A Mixed Integer Linear Programming model for systematically combining under-utilized sectors is described and its performance is compared to another recently-developed greedy heuristic algorithm. The MILP model produces slightly better results than the greedy algorithm. Improvements are greater as the number of sectors considered for combining increases. While the improvements are modest, over time they become significant. Lengthy computation times associated with large data sets are addressed by using the greedy algorithm as a starting solution for the optimal solver.

I. Introduction

Despite predictions of increased future air traffic congestion, many portions of the national airspace are, and will continue to be under-utilized. In fact, as shown in Ref. 1, 75% of the sectors in Cleveland center operate near or below half their capacity even during the busiest times of the day. When sectors with less traffic volume are combined into larger sector *clusters* for some length of time, fewer controller teams are required to manage the same airspace. By flexibly increasing or decreasing the number of sector clusters in response to air traffic volume during the course of a 24-hour period, resources at the Air Route Traffic Control Center can be allocated more efficiently.

Sector combining within centers is currently done today based on historical experience and operational policy. Refs. 1 and 2 present a method of performing the same operation on a systematic basis using a greedy algorithm that combines sectors into sector clusters on an hourly basis. The algorithm has the benefit of being computationally fast and flexible and is shown in simulation to produce fewer open sectors (or clusters of sectors) when compared with historical operational data of the same time period. However, this greedy algorithm may miss the optimal solution (the solution with the fewest number of sector clusters), especially as the number of sectors considered for combining grows. In Ref. 3 a hybrid branch-and-bound/neural network trained on a metric of controller workload is used to determine optimal sector combinations. However, it is presented with the intent of forecasting airspace configurations and traffic congestion—not as a method of tactical decision support in the vein of Refs. 1 and 2.

This paper presents a Mixed Integer Linear Programming model that finds an optimal solution of combined sectors resulting in the fewest number of sector clusters given current demand and capacity constraints. While the objective and constraints are nearly identical to that of Refs. 1 and 2, it is shown that this model provides a minor performance improvement over the greedy algorithm. Assuming sector combining is calculated for every hour in an operational setting, over the course of a day this minor improvement translates into a significant reduction in the number and duration of open sector clusters (measured as *sector-hours*). The improvement is greater as the number of sectors considered for combining increases. For instance, if a larger set of sectors is considered for combining than what is presently permitted, the greedy algorithm is more likely to miss the optimal cluster combination that produces the fewest number of clusters.

This paper is arranged as follows: Section II provides some background on present-day operations with additional details on the sector combining problem. Section III presents the approach used in developing the optimization model including all the necessary mathematical constraints. Section IV discusses how this system is implemented in software and the assumptions used in the simulation. Section V presents the

*Research Analyst, Systems Modeling and Optimization Branch, MS 210-8, NASA Ames Research Center, Moffett Field, CA 94035.

Copyright © 2009 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

results and compares the performance of the optimal Mixed Integer Linear Programming model to that of the greedy algorithm. Conclusions and directions of future work are discussed in Section VI.

II. Background

The National Airspace System (NAS) over the continental U.S. is partitioned into 20 Air Route Traffic Control Centers (ARTCCs, or *centers* for short). Each center is further subdivided into several smaller partitions known as *sectors*. Sectors are partitions in altitude as well as latitude and longitude, and they are typically classified as either low, high, or super-high according to their altitude range. Figure 1 shows the top view of Cleveland center. There are 27 high sectors, but only 11 are directly visible from above.

Each sector in the center is usually monitored by 1—3 air traffic controllers who communicate with the pilots of the aircraft in the sector to ensure safe and efficient traffic flow therein. However, during periods of low traffic volume, it is not always necessary to have a dedicated controller for every sector in the center. It often makes sense to combine two or more sectors together to increase the operational efficiency of the control center. In present-day operations, sectors are only permitted to combine with others in the same group of sectors known as an *area of specialization*. Each center contains several areas of specialization, which are contiguous subsets of sectors, and controllers are trained and certified to manage each of the sectors in their area. Thus, a single controller is capable of handling a cluster of combined sectors within his or her area of specialization when traffic permits. A simplified notion of three areas of specialization are represented by the three color shades in figure 1. Current research in sector design and the *generic sectors*[4] concept may, in the future, yield areas of specialization with more sectors, or eliminate their need all together.

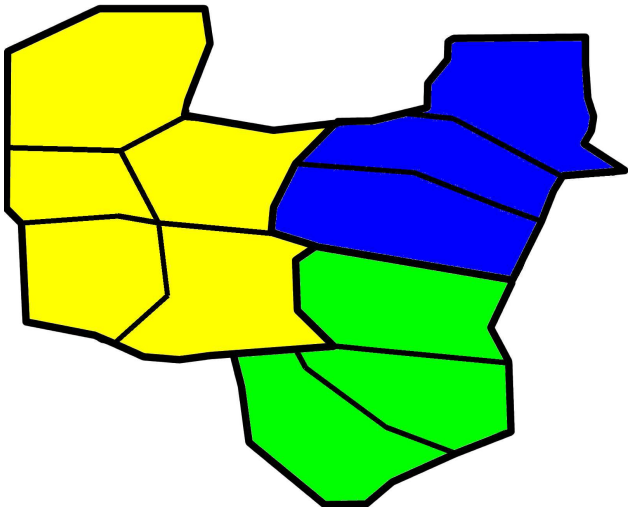


Figure 1. Top view of Cleveland Center. Colors represent a simplified notion of sector areas of specialization.

In current operations, the traffic demand and complexity within a sector must be within the controller team's ability to manage it. While various metrics of traffic complexity and sector capacity are currently being researched, the sector traffic complexity metric in operational use today is the maximum instantaneous aircraft count over a 15-minute interval.[1] This metric is forecast in multiple 15-minute intervals for some time horizon (typically one hour) and is available to controllers and their supervisors via ETMS[a] so they can anticipate future workload and make adjustments as necessary. The capacity of a sector is estimated by the Monitor Alert Parameter (MAP), which is the maximum instantaneous aircraft count suggested for a sector. Typically, controllers and traffic managers strive to keep the maximum instantaneous aircraft count of a sector less than the sector's MAP value, but violations of this constraint are not uncommon depending on traffic characteristics. Because these two metrics of sector demand and capacity are used in current air traffic operations, they are used in Ref. 1, as well as in this work for systematically combining sectors.

III. Approach

Similar to the approach in Refs. 1 and 2, the objective is to reduce the number of open sector clusters within a center with the condition that the capacity of each cluster cannot be exceeded. A cluster's demand over a future 15-minute interval is estimated by summing over all the sectors in the cluster the maximum instantaneous aircraft count of each sector. For a future interval of one hour, the largest of these values over the four future 15-minute intervals is taken to be the cluster's future one hour demand. The cluster capacity is conservatively estimated to be the maximum MAP value of all the sectors in the cluster. Finally, because the future demand for a cluster is based on sums of predicted peak aircraft counts, the estimated capacity

[a]Enhanced Traffic Management System: A software tool used by controllers to predict future traffic loads in sectors.

of the cluster is reduced by a safety cushion known as the *gap* parameter which accounts for errors due to traffic prediction uncertainties.

III.A. The Mixed Integer Linear Programming Model

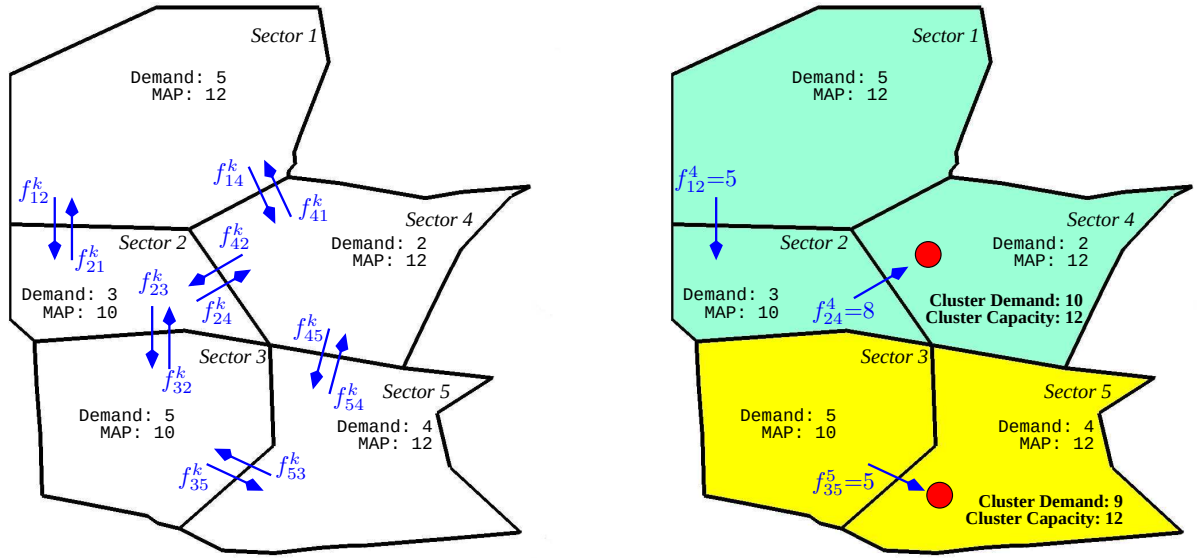
While both the objective and constraints of this problem are simple, implementing a technique to find the optimal solution is not. One difficulty lies in the size of the solution space. For instance, with as few as ten sectors, the number of possible cluster arrangements is nearly 116,000.[5] Although in reality, spatial contiguity would preclude many of these sector combinations from consideration, it is clear that solving the problem is not trivial due to the combinatorial size of the solution space which grows exponentially with the number of sectors. The greedy heuristic approach described in Ref. 1 is very efficient, but, as demonstrated herein, occasionally misses the optimal solution. On the other hand, the Mixed Integer Linear Program method promises optimality, but has its own complications.

Posing the clustering problem as a Mixed Integer Linear Programming (MILP) model that captures any feasible cluster's capacity and demand is not straightforward. The inspiration for this model comes from the optimal sector design method discussed in Refs. 6 and 7. This method makes use of a network flow model in which the sectors (nodes) are clustered by a connection (flow) variable passed between them. Connection *flow* is an abstract variable that is used to cluster sectors as it captures and sums up an attribute within them. Each sector is considered either a flow *source* or *sink*. In this model, the connection flow variable accumulates the cluster's demand by capturing the maximum number of predicted aircraft in a sector over a discrete time step interval and passing it from one sector to another, forming a sector cluster. Exactly one sector in the cluster must become a sink sector to absorb and terminate the flow. In this way, clusters are formed from the contiguous collection of sectors through which the flow is made to traverse, and the total predicted aircraft in a cluster is captured by the quantity of flow absorbed by the cluster's sink sector.

Sector traffic demands are predicted in terms of the maximum number of aircraft in a future 15-minute interval, so the maximum total demand of a given cluster of sectors must be computed accordingly. The predicted maximum peak aircraft count in a cluster over a 15-minute interval is the sum of the maximum peak counts of each sector in the cluster for this interval. Thus, if a sector cluster is to be operational for one hour, the greatest of these values over the four future 15-minute intervals yields the predicted maximum cluster demand over the next hour.

Figure 2 depicts a simplified example where only one time step is considered and only a subset of the sectors of Cleveland Center are considered. In figure 2(a) the estimated demand and capacity (MAP) values are shown for each sector along with all the possible flow paths between them. A viable cluster solution is depicted in figure 2(b). The details of the individual flow variables f_{ij}^k will be discussed in the next section. Note that the flow leaving sector 2 is equal to the flow going into sector 2 plus the traffic demand of sector 2. This flow path is then terminated in sector 4, which was selected to be the sink of that 3-sector cluster. Sectors 1 and 2 are source sectors. Clusters are identified by the index of their sink sector. Here the cluster demand is determined as the sum of the flow going into sector 4 plus the demand of sector 4 itself. Note that sink sectors also behave like source sectors in that they also contribute to the flow variable, but sink sectors, unlike source sectors, terminate flow by not permitting outward flow. Finally, the cluster capacity is set to the maximum MAP value of the cluster's sectors.

It must be stressed that there is no real-world meaning or implication to which sector in the cluster becomes a sink. Any sector is free to become a sink, because it must be possible to form "clusters" consisting of only one sector. The sink assignment is determined at run time *not* a priori. Furthermore, the concept of connection flow has no physical meaning in this formulation. That is, the path in which it is made to traverse has no relation to the direction traffic flows. In fact, in this setting it is fair to consider flow variables to be links that carry with them the predicted demand of each sector as they form an ordered set of sectors. The specific order of the sectors in a cluster as determined by these links is ultimately irrelevant. The primary advantage of this method is that contiguity of a sector cluster is guaranteed because it is easy to prevent flow (links) between non-adjacent sectors from being established. Contiguity is not a trivial constraint to enforce with other clustering methods. Here, it is inherent to the model structure. Also, by using the flow variable to sum up the traffic demand of the sectors in the cluster, the total cluster demand is easily captured, and circular flow path cycles are prevented because a conservation of flow in and out of each sector is enforced. The next section will detail how this model is implemented mathematically through MILP model decision variables and constraints.



(a) Unsolved set of connection flow variables f^k_{ij}.

(b) Clustered solution of flow variables and sink locations. Only non-zero values shown. Sink sectors indicated with red dot.

Figure 2. A notion of the network flow clustering model. Note the unsolved variables f^k_{ij} which when solved along with a sink location forms 2 sector clusters.

III.B. The MILP Model Formulation

Given N sectors considered for possible clustering, $\mathbf{I} = \{1...N\}$ is the set of sector indices, and $\mathbf{T} = \{1, 2, 3, 4\}$ is the set of discrete time steps for the four future 15-minute time intervals of the next hour, the MILP model parameters, decision variables, objective function, and constraints are now presented. Here, $i, j, k \in \mathbf{I}$, and $t \in \mathbf{T}$. Although indices i, j, and k all span the same set of sectors, index k is associated with the *cluster index* which is equal to the index of that cluster's sink sector. Recall that any sector is capable of becoming a sink for its cluster's connection flow, but many sectors will not become sinks. Thus, variables with k indices of non-sink sectors must be constrained to be zero-valued because they represent variables associated with non-existent clusters.

III.B.1. Model Parameters

The model parameters that are known a priori are

$\boldsymbol{AC_{it}}$ - the number of aircraft predicted to be in sector i at time step t,

$\boldsymbol{MAP_i}$ - the Monitor Alert Parameter for sector i,

$\boldsymbol{n_{ij} \in \{0, 1\}}$ - if sector i is a spatial neighbor of sector j, $n_{ij} = 1$, otherwise $n_{ij} = 0$, and

$\boldsymbol{gap_i \geq 0}$ - the gap parameter, a safety cushion on the predicted demand of a sector. This is usually identical for all sectors $i \in \mathbf{I}$, but may be individually altered for specific sector capacity violations.

III.B.2. Model Decision Variables

Decision variables are those that are not known a priori and are solved by the optimizing software at run time. The basic set of decision variables is

$\boldsymbol{f^k_{ij}} \in \mathbb{Z}^+$ - the flow of accumulated predicted aircraft count at time step $t = 1$ going from sector i to sector j and ultimately terminating at sink sector (cluster index) k,

$\boldsymbol{x_{ik} \in \{0, 1\}}$ - if sector i is assigned to cluster k, $x_{ik} = 1$, otherwise $x_{ik} = 0$,

$\boldsymbol{d_{kt}} \in \mathbb{Z}^+$ - the total air traffic demand of cluster k at time step t,

$\boldsymbol{D_k} \in \mathbb{Z}^+$ - the maximum demand of cluster k over all time steps $t \in \mathbf{T}$,

$\boldsymbol{C_k} \in \mathbb{Z}^+$ - the capacity of sector cluster k, and

$\boldsymbol{y_i} \in \{\mathbf{0}, \mathbf{1}\}$ - if sector i becomes a sink, $y_i = 1$, otherwise $y_i = 0$.

III.B.3. Objective Function and Constraints

The objective is to minimize the number of sector clusters, or in model terms, minimize the number of sectors that become sinks. Thus, the objective function is simply

$$\text{minimize} \sum_{i \in \mathbf{I}} y_i. \tag{1}$$

The primary constraint of preventing the traffic demand of each cluster from surpassing its capacity less the gap parameter is enforced by:

$$D_k \leq C_k - y_k \cdot gap_k \quad \forall k \in \mathbf{I} \tag{2}$$

The difficulty lies in implementing constraints that capture values for D_k and C_k. Implicit in these values is the assignment of sectors to clusters which requires determining values for x_{ik}. This, in turn, requires clustering the sectors according to the flow connection variables that are determined through the following constraints:

$$\sum_{j \in \mathbf{I}} f_{ij}^k = \sum_{j \in \mathbf{I}} f_{ji}^k + x_{ik} \cdot AC_{i1} \quad \forall i, k \in \mathbf{I} |\ i \neq k \tag{3}$$

$$d_{k1} = \sum_{j \in \mathbf{I}} f_{jk}^k + x_{kk} \cdot AC_{k1} \quad \forall k \in \mathbf{I} \tag{4}$$

$$\sum_{t \in \mathbf{T}} \frac{d_{kt}}{L} \leq y_k \quad \forall k \in \mathbf{I} \tag{5}$$

Constraint (3) sets up the conservation of flow among all source (non-sink) sectors. Considering flow heading to sink sector k (or, cluster k), the total flow out of sector i into all other sectors j must equal the total flow into sector i from all other sectors plus the number of predicted aircraft in sector i at time step $t = 1$. Note that the connection flow variable is based on traffic flow predicted for the first time step horizon AC_{i1}. This is because the flow only needs to operate on one additive sector attribute; the choice is arbitrary. The traffic demand for the remaining time steps is taken into consideration using an additional relation discussed below. Also, note that the assignment variables x_{ik} are used as a filter so that the flow variables for sectors i that are not assigned to cluster k are zero.

Constraint (4) is similar to constraint (3), but is specific for sectors that become sinks. Each sink sector k absorbs the flow heading to itself—there is no out-going flow. Thus, the demand variable d_{k1} is equal to the total flow into sector k based on traffic predictions for time $t = 1$ plus the number of predicted aircraft in sector k itself at time $t = 1$. This is the mechanism by which the total number of predicted aircraft in a cluster of sectors is captured. Again, refer to figure 2 for a depiction of this.

Because k spans all sectors $\mathbf{I}$ (since all sectors are possible sinks), d_{kt} must be zero for all non-sink sectors. This is accomplished by constraint (5). Here, and throughout the remainder of this section, L represents an arbitrary number sufficiently large enough to prevent the quotient on the left from being greater than one. Constraints of this form force the non-binary variable on the left to be zero when the binary variable on the right is zero, but if the binary variable is equal to one, the non-binary variable can be any value greater or equal to zero.

The next constraints define the relations between the connection flow variables f_{ij}^k, the sector-cluster assignment variables x_{ik}, and the sector-sink variables y_k.

$$\sum_{k \in \mathbf{I}} x_{ik} = 1 \quad \forall i \in \mathbf{I} \tag{6}$$

$$\sum_{j \in \mathbf{I}} \frac{f_{ij}^k}{L} \leq x_{ik} \quad \forall i, k \in \mathbf{I} \tag{7}$$

$$x_{kk} = y_k \quad \forall k \in \mathbf{I} \tag{8}$$

Constraint (6) forces every sector i to be assigned to exactly one cluster k. Constraint (7) is what relates the connection flow variable to an actual sector-cluster assignment. Each sector i is assigned to the same cluster k as the flow that leaves it. Because flow does not leave sink sectors, constraint (8) is necessary to define sectors assigned to themselves as sinks.

One of the primary reasons for using the network connection flow technique is that the contiguity of the clusters is guaranteed through the following simple constraint:

$$\sum_{k \in \mathbf{I}} \frac{f_{ij}^k}{L} \leq n_{ij} \quad \forall i, j \in \mathbf{I} \tag{9}$$

Constraint (9) restricts all flow from sector i from going into non-neighboring sectors as defined by the n_{ij} parameters. Thus, only sectors that are spatially contiguous may be clustered.

In addition to enforcing contiguity, the connection flow technique also yields cluster traffic demand for the first time step d_{k1} in constraint (4). Using the sector-cluster assignment variables x_{ik}, the remainder of the cluster demand variables d_{kt} for $t > 1$ can be related through the following equation:

$$d_{kt} = \sum_{i \in \mathbf{I}} x_{ik} \cdot AC_{it} \quad \forall k \in \mathbf{I}, t \in \mathbf{T} |\ t > 1 \tag{10}$$

Here, the x_{ik} variables serve as a filter in (10) so that the traffic demands of every cluster for the remaining time steps are the sum of predicted aircraft counts in only those sectors belonging to cluster k.

Finally, the cluster demand and capacity variables are given by:

$$D_k = \max_{t \in \mathbf{T}} \{d_{kt}\} \quad \forall k \in \mathbf{I} \tag{11}$$

$$C_k = \max_{i} \{x_{ik} \cdot \mathrm{MAP}_i\} \quad \forall k \in \mathbf{I} \tag{12}$$

Thus, constraint (2) can be realized. Recall that since k spans all sectors, for values of k involving non-sink sectors D_k, C_k, and y_k are zero so constraint (2) is never violated. Since constraints (11) and (12) invoke the *max* operator which is nonlinear they cannot be directly implemented. Fortunately, this is overcome with some additional constraints and variables discussed in the following section.

III.B.4. Linearizing Max Constraints

The technique for linearizing a *max* operator is well known, but will be presented here for thoroughness. Instead of implementing constraints (11) and (12) directly, each one is replaced by three linear constraints and the addition of a set of binary decision variables. Introducing $a_{kt} \in \{0, 1\}$, constraint (11) is enforced with the following:

$$D_k \geq d_{kt} \quad \forall k \in \mathbf{I}, t \in \mathbf{T} \tag{11a}$$

$$\frac{D_k - d_{kt}}{L} \leq 1 - a_{kt} \quad \forall k \in \mathbf{I}, t \in \mathbf{T} \tag{11b}$$

$$\sum_{t \in \mathbf{T}} a_{kt} = 1 \quad \forall k \in \mathbf{I} \tag{11c}$$

The strategy is that constraint (11a) bounds D_k on the bottom by the largest value of d_{kt}. Then constraint (11b) prevents D_k from being any larger than $\max_t\{d_{kt}\}$ by ensuring that the difference between D_k and d_{kt} is zero at least once since, by constraint (11c), for every k, $a_{kt} = 1$ exactly once across the t domain.

Likewise, with $b_{ik} \in \{0, 1\}$, nonlinear constraint (12) is implemented by the following linear constraints:

$$C_k \geq x_{ik} \cdot \mathrm{MAP}_i \quad \forall i, k \in \mathbf{I} \tag{12a}$$

$$\frac{C_k - x_{ik} \cdot \mathrm{MAP}_i}{L} \leq 1 - b_{ik} \quad \forall i, k \in \mathbf{I} \tag{12b}$$

$$\sum_{i \in \mathbf{I}} b_{ik} = 1 \quad \forall k \in \mathbf{I} \tag{12c}$$

Once again, L is a number large enough to prevent the quotient from being greater than 1.

III.C. Model Discussion

Given the above mathematical formulation, some observations are made about this MILP model. Referring again to figure 2, it is clear that the same cluster solution could be produced from different sink assignments as well as different flow arrangements. There are three possible sink locations for the top cluster, but there are also three different possible flow path arrangements for every possible sink location. (For example, flow could go from sector 2 to sector 1, then into sector 4 with $f_{21}^4 = 3$, and $f_{14}^4 = 8$.) Coupled with the two possible sink choices for the lower cluster (with only one flow arrangement possible for each), there are exactly eighteen unique MILP model solutions that produce the exact same sector clustering arrangement shown.

Furthermore, depending on traffic demand and sector capacity, there may be more than one feasible cluster arrangement that satisfies constraints and results in the fewest number of clusters. Even the greedy heuristic approach could, in theory, produce alternate clustering solutions having the same number of clusters. After all, it is only the *number* of clusters that matters to the objective function of both methods, not the specific arrangement. The optimal MILP model, however, can produce exponentially more internal variable specific solutions (distinguished only by flow and sink variables) than the greedy heuristic approach that result in the same number of clusters. Thus, the network flow model adds a significant level of complexity that can lead to lengthy computation times. This will be discussed in greater detail in Section V. The implementation of the model in simulation is presented next.

IV. Implementation

Based on historical flight data, the Future ATM Concepts Evaluation Tool (FACET)[8] is used to record the sector traffic loading of a center. In practice only predicted traffic can be used to make decisions on sector combining, but for the simulations herein, actual traffic loads are used for the four 15-minute look-ahead time steps. Thus, given a specific center, data for AC_{it} is recorded for each sector i at four time steps for each hour.

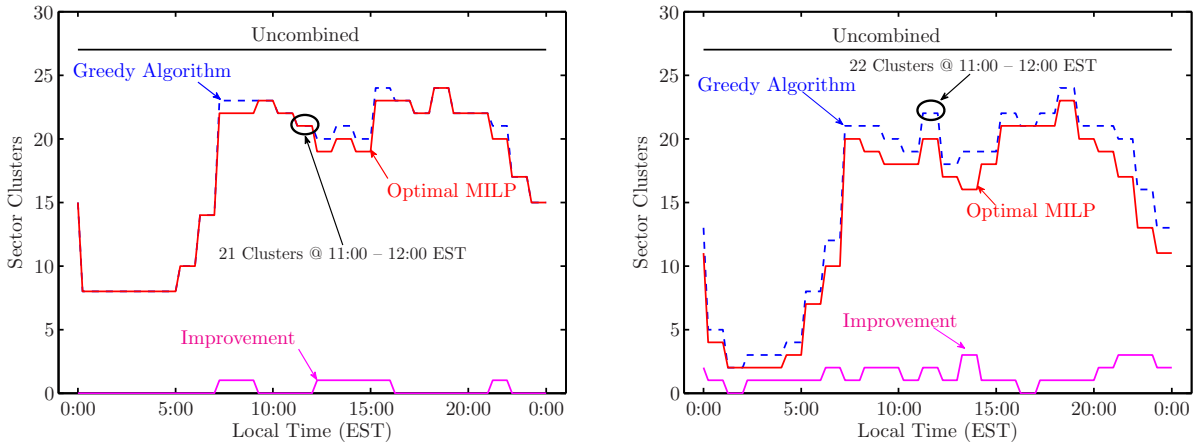
For every sector i in the center, its MAP value MAP_i, as well as its set of spatial neighbors n_{ij} is determined. The default gap parameter is set at three, so $gap_i = 3 \ \forall i \in \mathbf{I}$, except for the case when traffic in an individual sector violates the capacity constraint. Thus, if for some sector i the peak traffic in the next hour AC_{it} exceeds $MAP_i - 3$, the gap parameter for that sector is lowered so that constraint (2) can be met. This is necessary because without it, the MILP model is infeasible. By reducing gap_i to the point that constraint (2) is just barely met means that the sector in question will not be combined with any other sector, and will become a cluster of one. Recall that the intent of the *gap* parameter is to be a cushion for traffic prediction error. Once the model parameters are captured from FACET, the model is solved using either the open-source GNU Linear Programming Kit (GLPK)[9] optimization package, or the commercial AMPL/CPLEX[10] package.

V. Results

For these simulations, historical flight data from 8 February 2007 in Cleveland Center are used to compare the performance of the optimal MILP sector combining method to that of the greedy algorithm method. Only high sectors are considered for combining. In the first scenario, sectors are only permitted to combine with those in the same area of specialization. This is known as the *restricted* case, and the total number of resulting high-sector clusters are shown in figure 3(a) for each hour over a 24-hour period. There are 27 high-altitude sectors in this center grouped into eight areas of specialization. This means that at each time step eight separate optimal solutions are found—one for each area of specialization.

The results demonstrate that the greedy algorithm produces near-optimal results. The MILP model provides, at most, an improvement of one less cluster for some time steps. However, given the relatively small size of each area, it is not surprising that improvements are modest. Nevertheless, over the course of this day, a savings of seven sector-hours is realized. (A *sector-hour* refers to one sector, or cluster of combined sectors operational for one hour.) Over several weeks, this savings adds up and becomes significant.

As the number of sectors considered for combining increases, intuition suggests that because the solution space increases exponentially, the greedy algorithm is more likely to miss the optimal combined sector solution. This assumption is validated by testing the performance of both methods, while allowing all the



(a) Sector combinations are restricted to areas of specialization. (b) Sector combinations are unrestricted by areas of specialization.

Figure 3. Total number of high sector clusters produced by the optimal MILP method compared to the greedy algorithm method. Based on traffic data for 8 February 2007 in Cleveland Center.

high sectors to combine without regard to their areas of specialization. This is known as the *unrestricted* case, and the results are shown in figure 3(b). The same traffic data from Cleveland Center is used and all 27 sectors are allowed to combine without restriction. Here, the difference between the two methods is much more dramatic with a total improvement of 34 sector-hours for the day. In figure 4 the MILP results for 0:00 – 1:00 EST are shown. It is easy to see by the size and convoluted arrangement of the clusters why such unrestricted combinations are not performed in current operations. Nevertheless, the unrestricted scenario is a rigorous test of the performance of both methods' ability to handle greater numbers of sectors for combining. This will be important if the areas of specialization are enlarged, as would be possible among generic sectors. Also, while it is not investigated here, both methods could benefit by selecting solutions from a pool of optimal or near-optimal solutions according to additional criteria.

The tendency of the greedy algorithm to not only miss the optimal solution, but to produce inconsistent results depending on the input data is best demonstrated by the results at 11:00 – 12:00 EST. Here, note that in figure 3(a) the greedy algorithm produces a result of 21 sector clusters whereas in the unrestricted case, where the number of clusters should obviously be no greater than the restricted case, figure 3(b) shows it produces a solution of 22 sector clusters. Because the optimal MILP method is not subject to these types of inconsistencies, the unrestricted case always has equal or fewer sector clusters than the restricted one.

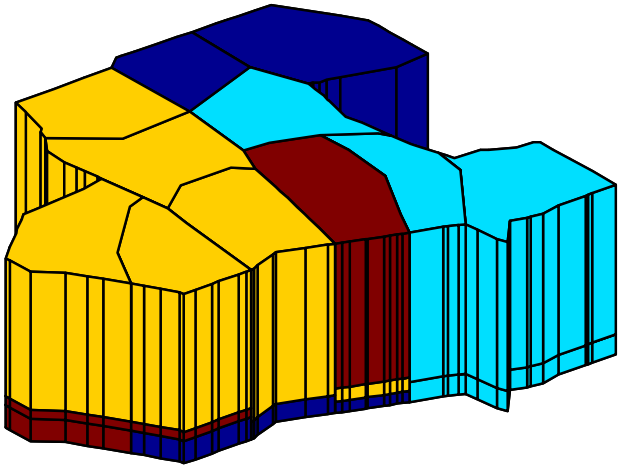


Figure 4. An example of unrestricted cluster results produced by the optimal MILP for 0:00 – 1:00 EST consisting of 4 sector clusters.

These results focus on the resulting number of clusters that each method produces, not the cluster arrangement themselves. It should be noted, that when the two methods produce the same number of clusters at a given time step, the clusters may or may not be identical. As discussed above, there may be more than one possible cluster arrangement depending on traffic conditions. Some arrangements may be more operationally practical than others, but no attempt to analyze them has been made here.

V.A. Computation Time

Though the MILP model is capable of finding the optimal solution to the sector combining problem, the computation times can be significant. For the restricted case when the size of the problem is limited to the number of sectors in an area of specialization (the largest of which contains six sectors), solution times range in the tens of seconds using the open-source GLPK package. When the number of sectors considered for clustering is increased to all 27 high sectors solution times for each time step are often measured in hours using the commercial (and computationally efficient) AMPL/CPLEX package. In practice, this would render the method useless, but fortunately the greedy algorithm (with computation times less than a minute) can be used to provide a starting solution for the optimal solver, thus dramatically reducing solution times. In fact, this technique was used for the unrestricted simulation case shown above. If time does not permit the exact optimal solution to be found, the solver will usually find a superior (if not optimal) solution within several minutes.

VI. Conclusions

This paper describes a method of optimally combining sectors into clusters according to traffic demand and capacity. The results demonstrate that the existing greedy algorithm is very efficient for quickly determining a near-optimal solution to the sector combining problem; as long as the problem size is small, it finds the optimal solution most of the time. With a small number of sectors, the improvement offered by the optimal MILP model is slight. However this improvement becomes significant over time.

When dealing with a large number of sectors, the optimal MILP model demonstrates a significant and consistent improvement over the greedy algorithm. However, this comes with the cost of exponentially increasing computation times. This is addressed by using the greedy algorithm to provide an initial solution for the optimal model, which drastically reduces computation times. Doing so allows the optimization software to find improved solutions in matters of minutes as opposed to hours.

While the optimal MILP model shows promise for improving a center's workload efficiency, unlike the greedy algorithm, it is not capable of handling nonlinear or non-additive sector workload metrics. An active area of research is investigating methods of more accurately estimating sector capacity and workload. Unlike the conservative methods used in this optimal MILP model and the existing greedy algorithm, many of the proposed methods (like Dynamic Density) involve nonlinear and/or non-additive metrics. Such metrics are difficult, if not impossible, to incorporate into the MILP domain. Research is needed to evaluate the accuracy of these alternative metrics, and to determine the extent of improvement they offer over the basic metrics of aircraft count and MAP when used in sector combining methods. While it may be impossible to directly apply these metrics to the MILP model, it is certainly possible to evaluate the many optimal or near-optimal solutions the MILP model produces according to any number of additional constraints. The desirable solution could then be selected accordingly.

Nevertheless, this optimal MILP model may be a useful tool for suggesting sector combinations in both present and future operations. Also, if future concepts like generic sectors are adopted, it may be feasible to combine larger sets of sectors without area of specialization restrictions. In such case, the optimal MILP is especially useful for determining the minimum feasible number of clusters given traffic predictions and sector capacity.

Acknowledgments

The author would like to thank Michael Bloem and Pramod Gupta for providing details of their sector combining algorithm along with their simulation results.

References

[1]Bloem, M. and Kopardekar, P., "Combining Airspace Sectors for the Efficient Use of Air Traffic Control Resources," *Proceedings of AIAA Guidance Navigation and Control Conference*, Honolulu, HI, August 2008.

[2]Bloem, M., Gupta, P., and Kopardekar, P., "Algorithms for Combining Airspace Sectors," *Air Traffic Control Quarterly*, 2009, Submitted.

[3]Gianazza, D., Allignol, C., and Saporito, N., "An Efficient Airspace Configuration Forecast," *Proc. of 8th USA/Europe ATM Seminar*, Napa, CA, USA, June 29 - July 2 2009.

[4]Kopardekar, P., Bilimoria, K. D., and Sridhar, B., "Airspace Configuration Concepts for the Next Generation Air Transportation System," *Air Traffic Control Quarterly*, Vol. 16, No. 3, 2008.

[5]Theodoridis, S. and Koutroumbas, K., *Pattern Recognition*, chap. 12, Amsterdam [etc.] : Elsevier, 3rd ed., 2006, pp. 517–518.

[6]Yousefi, A., Khorrami, B., Hoffman, R., and B.Hackney, "Enhanced Dynamic Airspace Configuration Algorithms and Concepts," Tech. Rep. Report No. 34N1207-001-R0, Metron Aviation Inc., 45300 Catalina Court, Suite 101, Dulles, VA 20166, December 2007.

[7]Drew, M., "Analysis of an Optimal Sector Design Method," *27th Digital Avionics Systems Conference*, St. Paul, MN, October 2008.

[8]Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K. S., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, Vol. 9, No. 1, 2001, pp. 1–20.

[9]"GLPK (GNU Linear Programming Kit)," July 2009, `http:/www.gnu.org/software/glpk/`.

[10]ILOG, Inc., "AMPL," July 2009, `http://www.ilog.com/products/ampl/`.